WHAT IS CLAIMED IS:

1. A method of maintaining an index during a reorganization of data in a database, wherein said database comprises one or more records and each said record includes a root segment, the method comprising:

storing each root segment in a separate storage location; and

retaining each root segment in its storage location during a reorganization of data in said database.

2. The method of claim 1, wherein all root segments stored within a block of storage locations are stored in contiguous storage locations.

3. The method of claim 1, wherein each said storage location which stores a root segment is fixed.

4. The method of claim 2, wherein each said storage location which stores a root segment is fixed.

5. The method of claim 1, wherein said database is an IMS full function database.

6. The method of claim 3, wherein each said root segment is stored in a fixed storage location at the time it is added to the database.

7. A method of maintaining an index during a reorganization of data in a database, wherein said database comprises one or more records and each said record includes a root segment and one or more non-root segments, the method comprising:

storing each root segment in a separate storage location, wherein each storage location is associated with a particular block of storage locations;

retaining each root segment in its storage location during a reorganization of data in said database; and

storing each non-root segment, associated with a first root segment, in a block of storage locations in which said first root segment is also stored.

8. The method of claim 7, wherein all root segments stored within a block of storage locations are stored in contiguous storage locations.

9. The method of claim 7, wherein each said storage location which stores a root segment is fixed.

10. The method of claim 8, wherein each said storage location which stores a root segment is fixed.

11. The method of claim 7, wherein said database is an IMS full function database.

12. The method of claim 9, wherein each said root segment is stored in a fixed storage location at the time it is added to the database.

13. A method of maintaining an index during a reorganization of data in a database, wherein said database comprises one or more records, each said record includes a root segment and one or more non-root segments, and each root segment comprises a prefix component and a data component, the method comprising:

storing said prefix component of each root segment in a separate storage location; and

retaining said prefix component of each root segment in its storage location during a reorganization of data in said database.

14. The method of claim 13, wherein all root segment prefix components stored within a block of storage locations are stored in contiguous storage locations.

15. The method of claim 13, wherein each said storage location which stores a root segment prefix component is fixed.

16. The method of claim 14, wherein each said storage location which stores a root segment prefix component is fixed.

17. The method of claim 13, wherein said database is an IMS full function database.

18. The method of claim 15, wherein each said root segment prefix component is stored in a fixed storage location at the time it is added to the database.

19. A method of maintaining an index during a reorganization of data in a database, wherein said database comprises one or more records, each said record includes a root segment and one or more non-root segments, and each root and non-root segment comprises a prefix component and a data component, the method comprising:

storing said prefix component of each root segment in a separate storage location, wherein each storage location is associated with a particular block of storage locations;

retaining said prefix component of each root segment in its storage location during a reorganization of data in said database; and

storing said prefix component of each non-root segment which is associated with a first root segment, in a block of storage locations in which said prefix component of said first root segment is also stored.

20. The method of claim 19, wherein all root segments stored within a block of storage locations are stored in contiguous storage locations.

21. The method of claim 19, wherein said database is an IMS full function database.

22. A method of facilitating correction of an index after a reorganization of data in a database, wherein said index comprises index entries, said database comprises one or more records, each said record comprises one or more segments, and each index entry comprises an address to a target segment, the method comprising:

prior to a reorganization of data in said database, assigning a unique token to each target segment and each corresponding index entry having an address to a target segment, wherein said unique token for a given target segment and for a corresponding index entry is the same;

after a reorganization of data in said database, reading the unique token of a first index entry;

reading the unique token of each target segment until a match is found between the unique token of a matching target segment and the unique token of said first index entry;

determining the address of said matching target segment; and

replacing the address of said first index entry with the address of said matching target segment.

23. The method of claim 19, wherein said index which is to be corrected is a secondary index and each said segment is a non-root segment.

24. The method of claim 22, wherein said database is an IMS database.

25. The method of claim 23, wherein said database is an IMS database.

26. A method of facilitating correction of an index after a reorganization of data in a database, wherein said index comprises index entries, said database comprises one or more records, each said record comprises a root segment and one or more non-root segments, and each index entry comprises an address to a target segment included within said root and non-root segments, the method comprising:

prior to a reorganization of data in said database, assigning a unique token to each target segment and each corresponding index entry having an address to a target segment, wherein said unique token for a given target segment and for a corresponding index entry is the same;

after a reorganization of data in said database, reading the unique token of a first index entry for a first record;

reading the unique token of each non-root segment within said first record until a match is found between the unique token of a matching target segment and the unique token of said first index entry;

determining the address of said matching target segment; and

replacing the address of said first index entry with the address of said matching target segment.

27. The method of claim 26, wherein said index which is to be corrected is a secondary index.

28. The method of claim 26, wherein said database is an IMS database.

29. The method of claim 27, wherein said database is an IMS database.

30. The method of claim 26, wherein each said unique token includes one or more the following: (i) a born on date of the target segment to which said unique token is assigned; or (ii) a key field of the target segment to which said unique token is assigned.

31. A method of facilitating correction of an index after a reorganization of data in a database, wherein said index comprises index entries, said database comprises one or more records, each said record comprises a root segment and one or more non-root segments, and each index entry comprises an address to a target segment included within said root and non-root segments, wherein said root segment and one or more non-root segments for a record are stored within a block of storage locations, the method comprising:

prior to a reorganization of data in said database, assigning a unique token to each target segment and each corresponding index entry having an address to a target segment, wherein said unique token for a given target segment and for a corresponding index entry is the same;

after a reorganization of data in said database, reading the unique token of a first index entry for a first record;

reading the unique token of each non-root segment within said first record until a match is found between the unique token of a matching target segment and the unique token of said first index entry;

determining the address of said matching target segment; and

replacing the address of said first index entry with the address of said matching target segment.

32. The method of claim 31, wherein said index which is to be corrected is a secondary index.

33. The method of claim 31, wherein said database is an IMS database.

34. The method of claim 32, wherein said database is an IMS database.

35. The method of claim 31, wherein each said root segment is stored in a fixed storage location prior to a reorganization of data in said database, and said root segment is retained in said fixed storage location during a reorganization.

36. The method of claim 31, wherein each said unique token is includes one or more the following: (i) a born on date of the target segment to which said unique token is assigned; or (ii) a key field of the target segment to which said unique token is assigned.

37. A method of facilitating correction of an index after a reorganization of data in a database, wherein said index comprises index entries, said database comprises one or more records, each said record comprises a root segment and one or more non-root segments, each root and non-root segment comprises a prefix component and a data component, and each index entry comprises an address to the prefix component of a target segment included within said root and non-root segments, the method comprising:

prior to a reorganization of data in said database, assigning a unique token to the prefix component of each target segment and each corresponding index entry having an address to the prefix component of a target segment, wherein said unique token for the prefix component of a given target segment and for a corresponding index entry is the same;

after a reorganization of data in said database, reading the unique token of a first index entry for a first record;

reading the unique token of the prefix component of each non-root segment within said first record until a match is found between the unique token of a matching target segment prefix component and the unique token of said first index entry;

determining the address of said matching target segment prefix component; and

replacing the address of said first index entry with the address of said matching target segment prefix component.

38. The method of claim 37, wherein said index which is to be corrected is a secondary index.

39. The method of claim 37, wherein said database is an IMS database.

40. The method of claim 37, wherein each said root segment is stored in a fixed storage location prior to a reorganization of data in said database, and said root segment is retained in said fixed storage location during a reorganization.

41. The method of claim 37, wherein each index entry and each non-root target segment further comprise a root segment identifier which identifies what root segment said non-root target segment is associated with, and each identified root segment comprises addresses to all non-root segments, within a record, associated with said identified root segment.

42. The method of claim 41, wherein each said unique token includes one or more the following: (i) a born on date of the target segment to which said unique token is assigned; or (ii) a key field of the target segment to which said unique token is assigned.

43. The method of claim 42, wherein each said unique token for an index entry and each non-root target segment further comprises said root segment identifier which identifies what root segment said non-root target segment is associated with.

44. The method of claim 41, wherein said database is an IMS database.

45. The method of claim 41, wherein said index which is to be corrected is a secondary index.

46. The method of claim 37, wherein each said unique token includes one or more the following: (i) a born on date of the target segment to which said unique token is assigned; or (ii) a key field of the target segment to which said unique token is assigned.

47. The method of claim 22, wherein after a reorganization of data in said database but before taking steps to correct said address of said first index entry, determining if said address of said first index entry is valid and then correcting said address only if it is invalid.

48. The method of claim 47, wherein determining if said address of said first index entry is valid comprises comparing the unique token of said first index

entry to the unique token associated with a segment located at said address, and ascertaining if said unique tokens are the same.

49. The method of claim 47, wherein determining if said address of said first index entry is valid comprises comparing a segment code of said first index entry to a segment code associated with a segment located at said address, and if said segment codes are the same, then comparing the unique token of said first index entry to the unique token associated with said segment located at said address, and ascertaining if said unique tokens are the same.

50. The method of claim 26, wherein after a reorganization of data in said database but before taking steps to correct said address of said first index entry, determining if said address of said first index entry is valid and then correcting said address only if it is invalid.

51. The method of claim 50, wherein determining if said address of said first index entry is valid comprises comparing the unique token of said first index entry to the unique token associated with a segment located at said address, and ascertaining if said unique tokens are the same.

52. The method of claim 50, wherein determining if said address of said first index entry is valid comprises comparing a segment code of said first index entry to a segment code associated with a segment located at said address, and if said segment codes are the same, then comparing the unique token of said first index entry to the unique token associated with said segment located at said address, and ascertaining if said unique tokens are the same.

53. The method of claim 31, wherein after a reorganization of data in said database but before taking steps to correct said address of said first index entry, determining if said address of said first index entry is valid and then correcting said address only if it is invalid.

54. The method of claim 53, wherein determining if said address of said first index entry is valid comprises comparing the unique token of said first index entry to the unique token associated with a segment located at said address, and ascertaining if said unique tokens are the same.

55. The method of claim 53, wherein determining if said address of said first index entry is valid comprises comparing a segment code of said first index entry to a segment code associated with a segment located at said address, and if said segment codes are the same, then comparing the unique token of said first index entry to the unique token associated with said segment located at said address, and ascertaining if said unique tokens are the same.

56. The method of claim 37, wherein after a reorganization of data in said database but before taking steps to correct said address of said first index entry, determining if said address of said first index entry is valid and then correcting said address only if it is invalid.

57. The method of claim 56, wherein determining if said address of said first index entry is valid comprises comparing the unique token of said first index entry to the unique token associated with a prefix component of a segment located at said address, and ascertaining if said unique tokens are the same.

58. The method of claim 56, wherein determining if said address of said first index entry is valid comprises comparing a segment code of said first index entry to a segment code associated with a prefix component of a segment located at said address, and if said segment codes are the same, then comparing the unique token of said first index entry to the unique token associated with said prefix component of a segment located at said address, and ascertaining if said unique tokens are the same.

59. A program storage media readable by a machine and containing instructions for performing the method contained in claim 1.

60. A program storage media readable by a machine and containing instructions for performing the method contained in claim 7.

61. A program storage media readable by a machine and containing instructions for performing the method contained in claim 13.

62. A program storage media readable by a machine and containing instructions for performing the method contained in claim 19.

63. A program storage media readable by a machine and containing instructions for performing the method contained in claim 22.

64. A program storage media readable by a machine and containing instructions for performing the method contained in claim 26.

65. A program storage media readable by a machine and containing instructions for performing the method contained in claim 31.

66. A program storage media readable by a machine and containing instructions for performing the method contained in claim 37.